

1 / 50

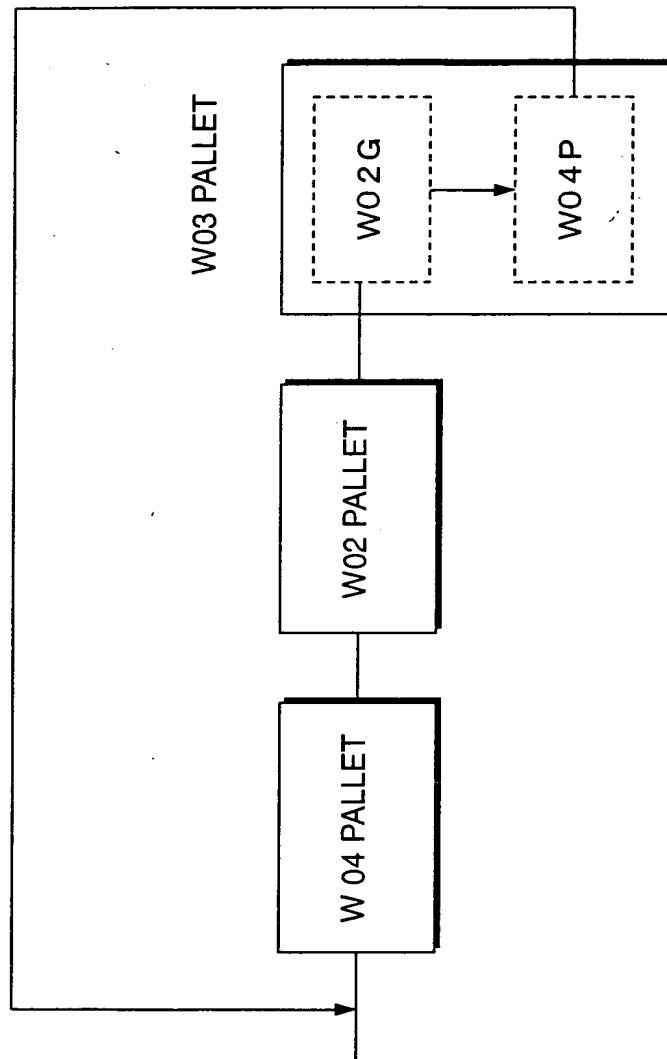
Fig. 1

Fig. 2

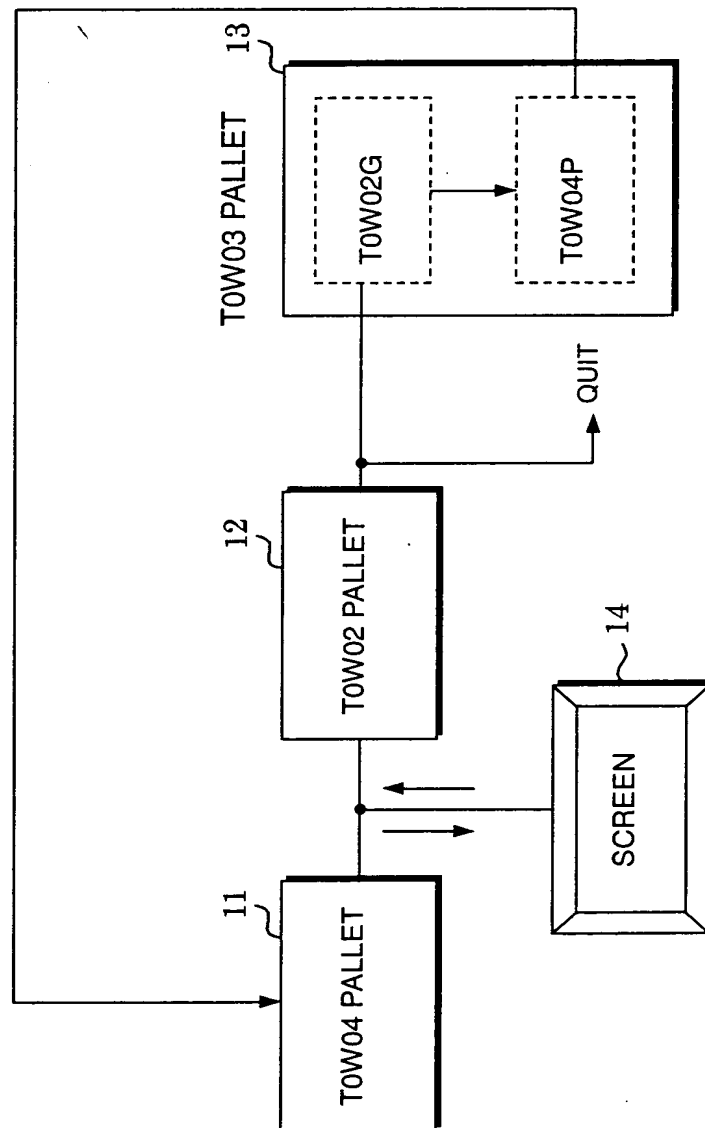


Fig. 3

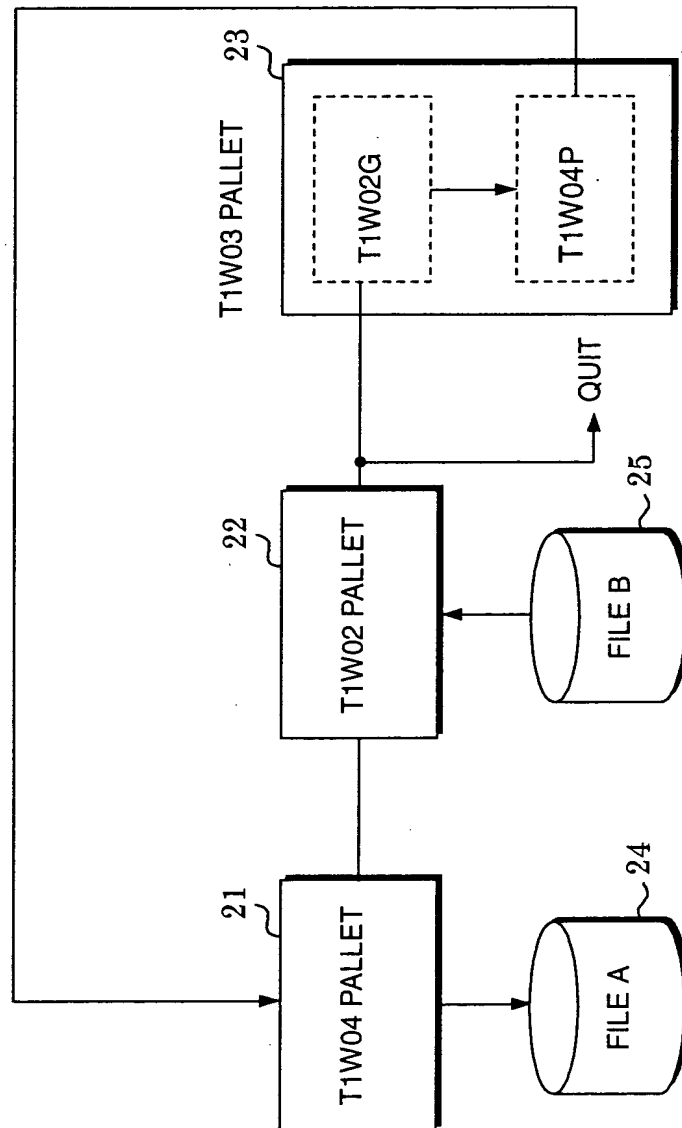
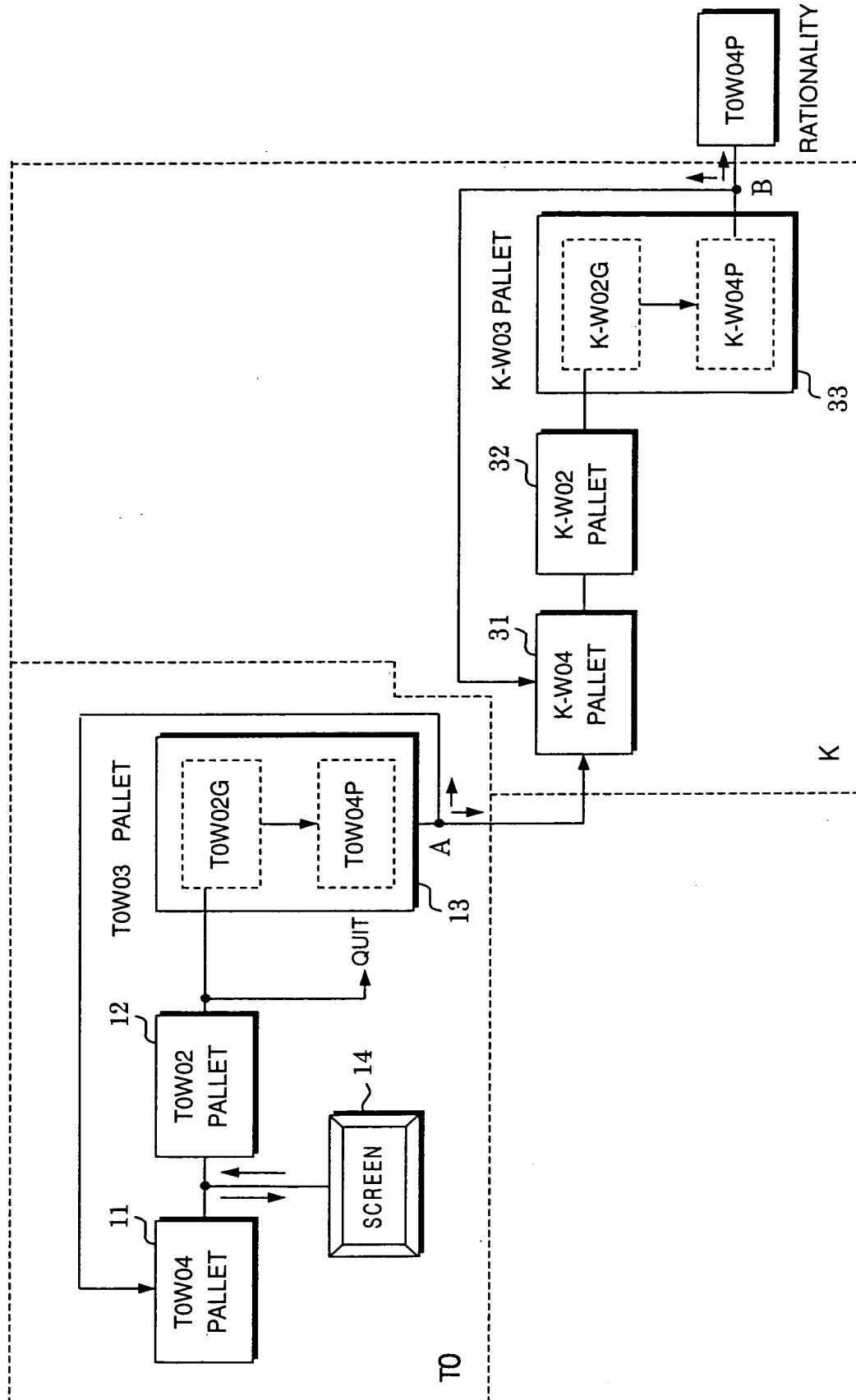


Fig. 4



5 / 50

Fig. 5

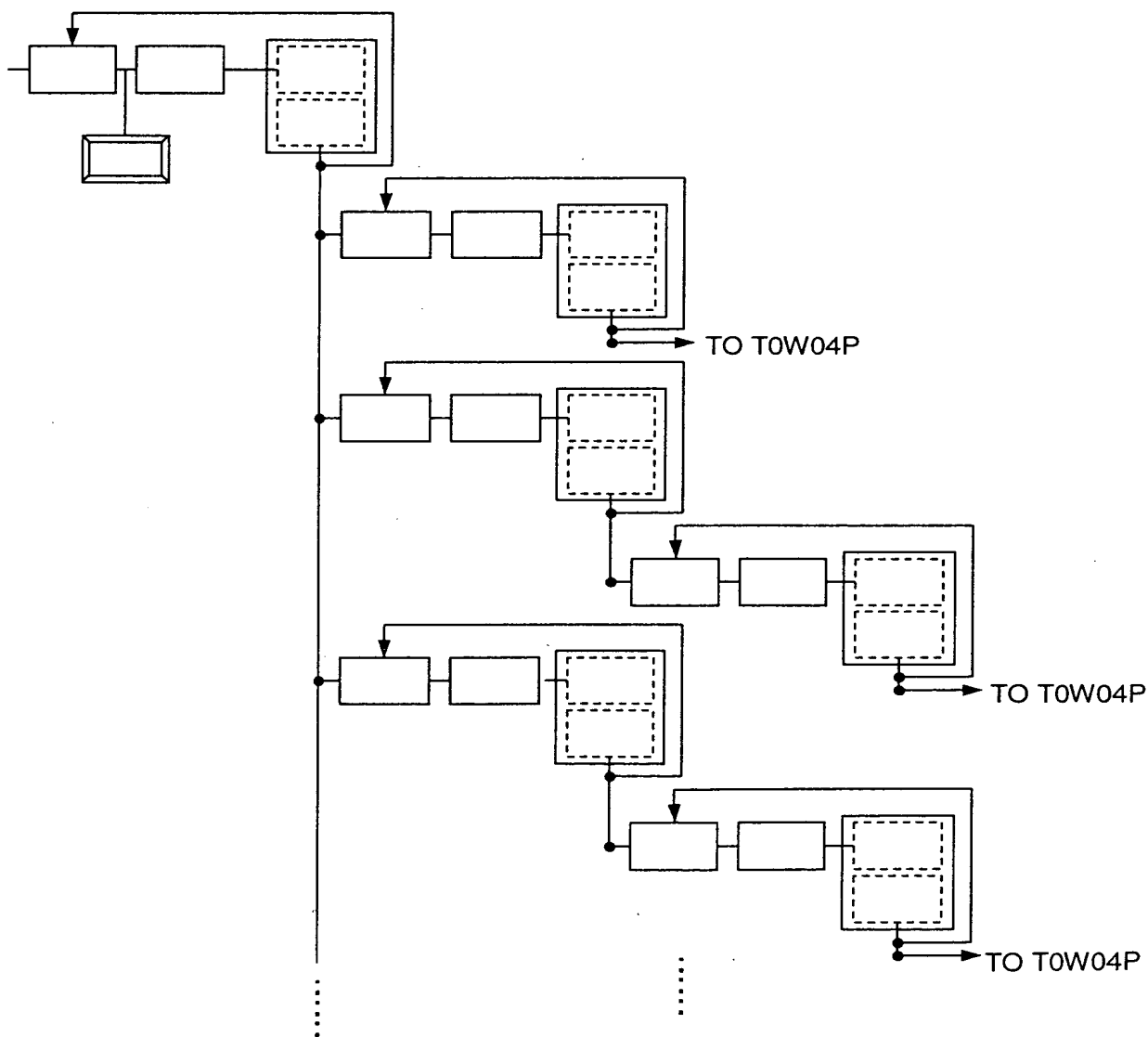


Fig. 6

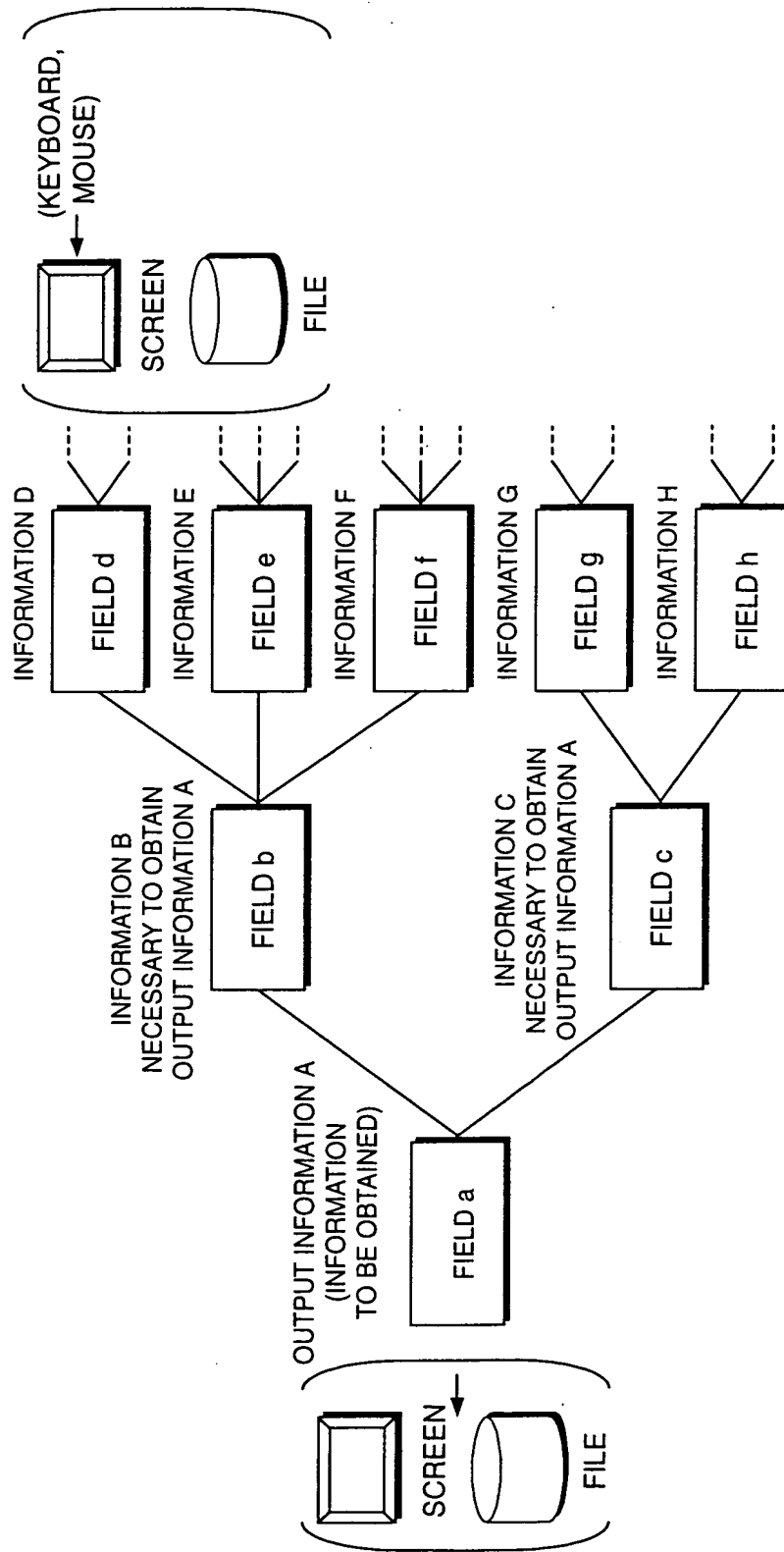
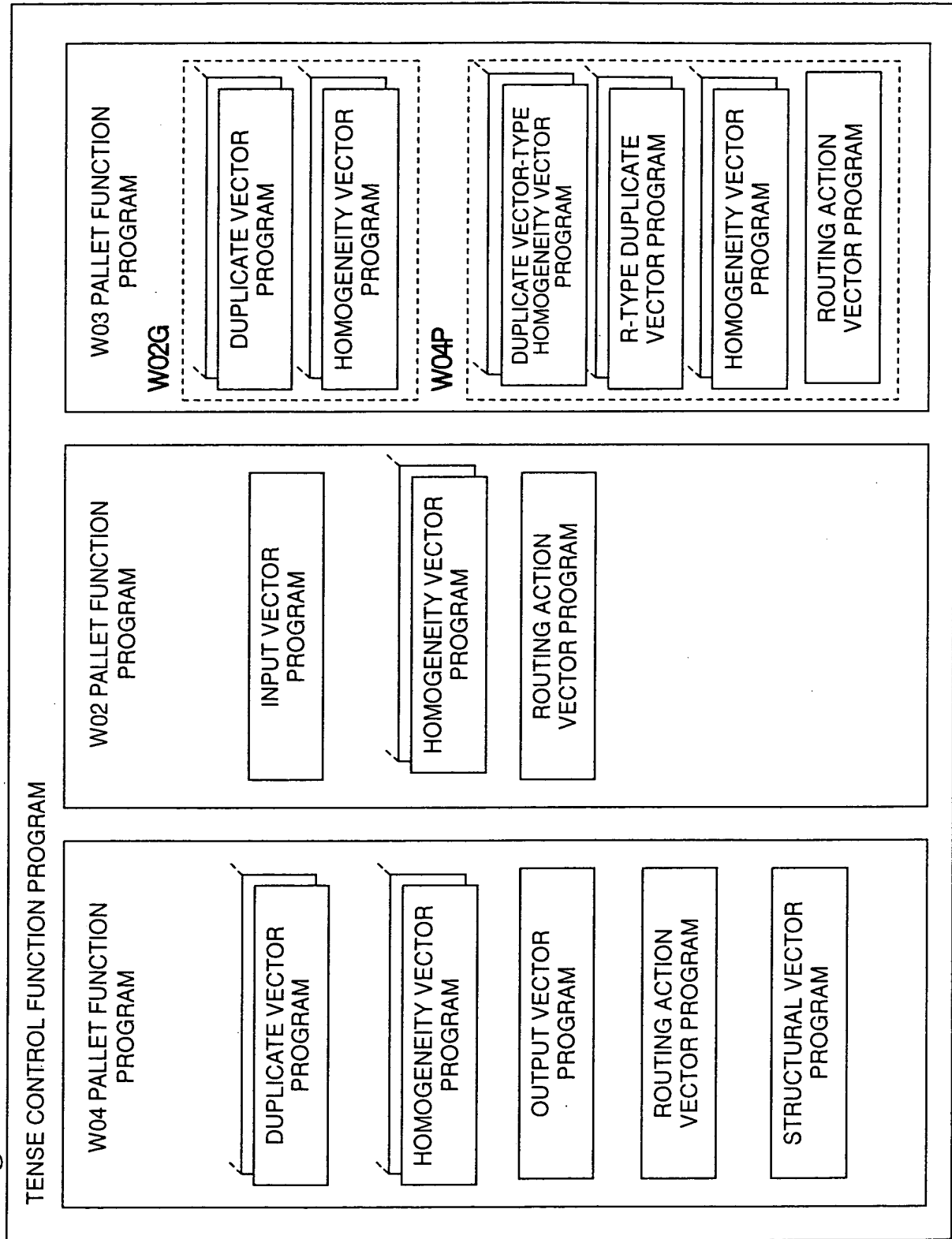


Fig. 7



W04 PALLET FUNCTION PROGRAM

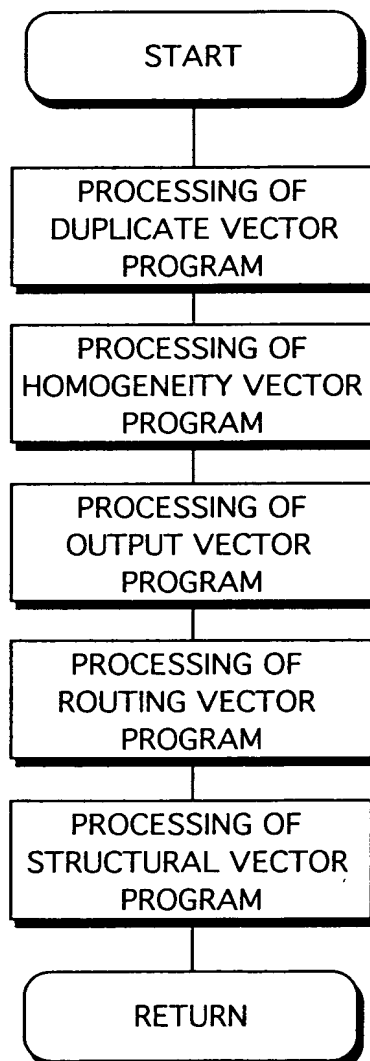
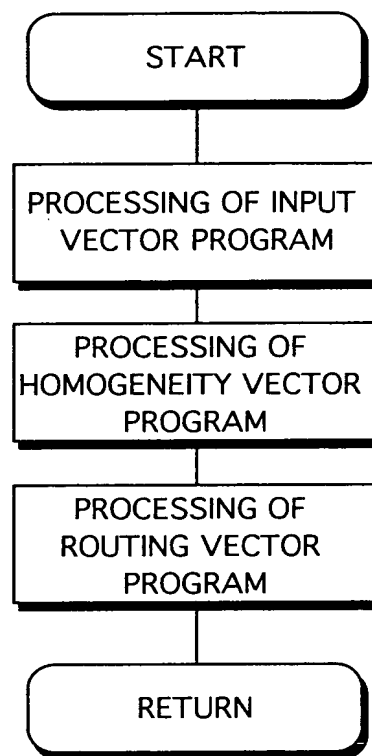


Fig. 9

W02 PALLET FUNCTION PROGRAM



09/868893

Fig. 10

W03 PALLET FUNCTION PROGRAM

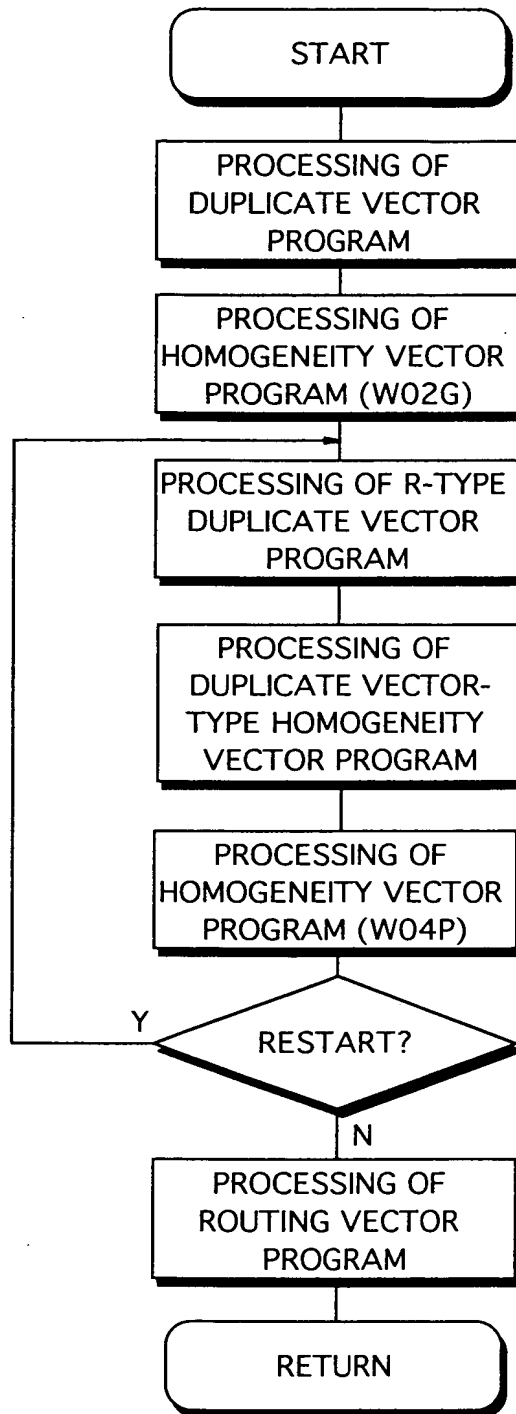


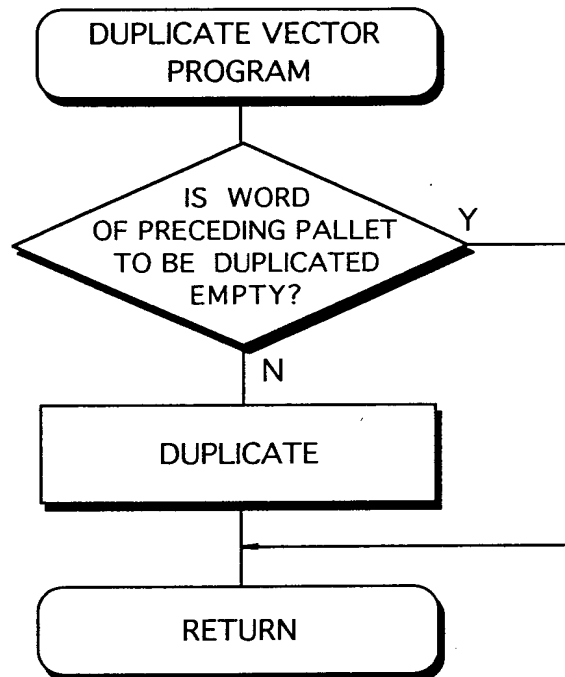
Fig. 11

Fig. 12

HOMOGENEITY VECTOR PROGRAM (W04)

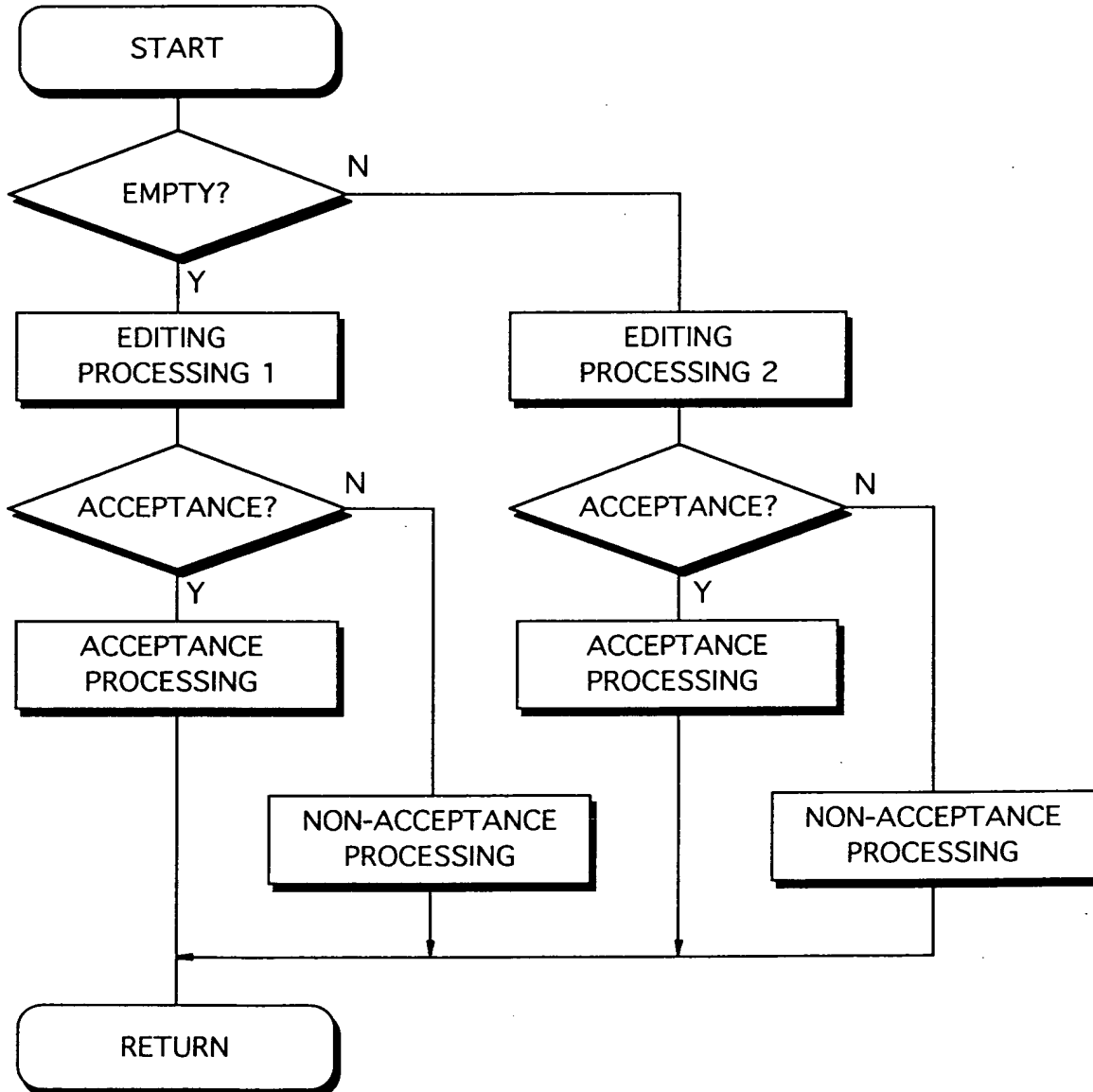


Fig. 13

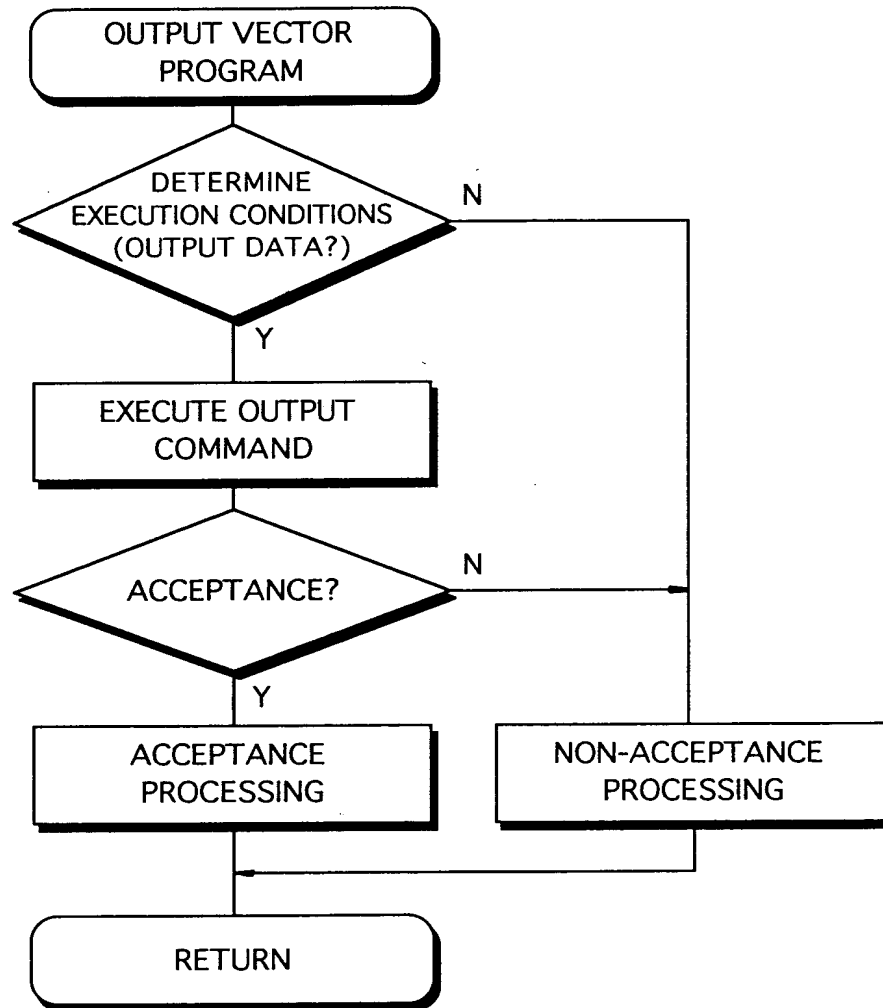
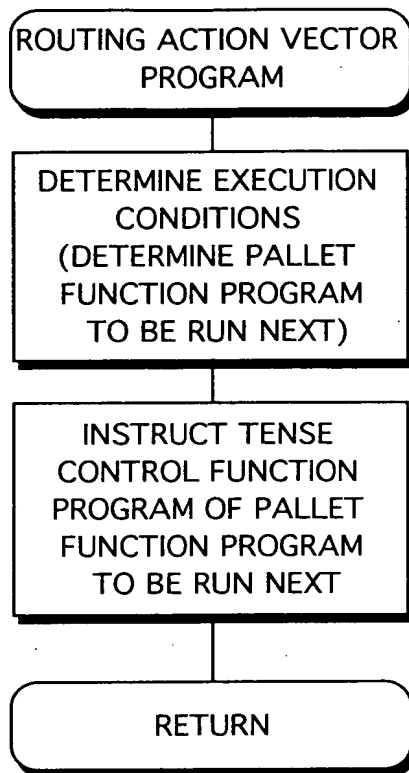


FIG. 13

Fig. 14

T03040" 6669360

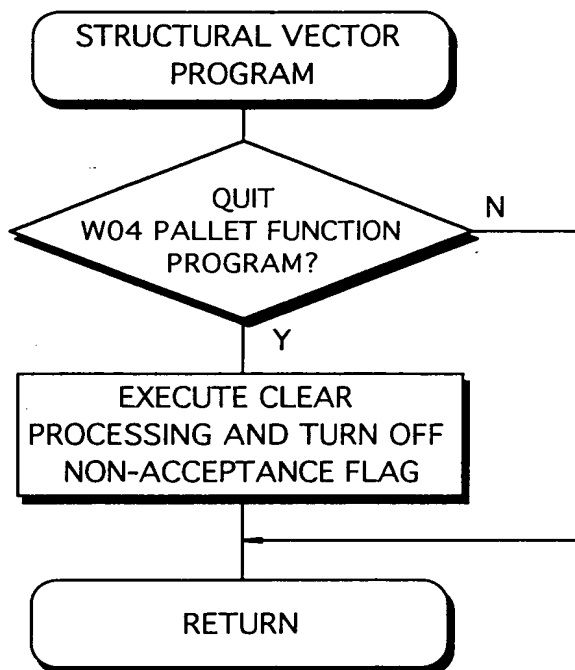
Fig. 15

Fig. 16

HOMOGENEITY VECTOR PROGRAM (W02)

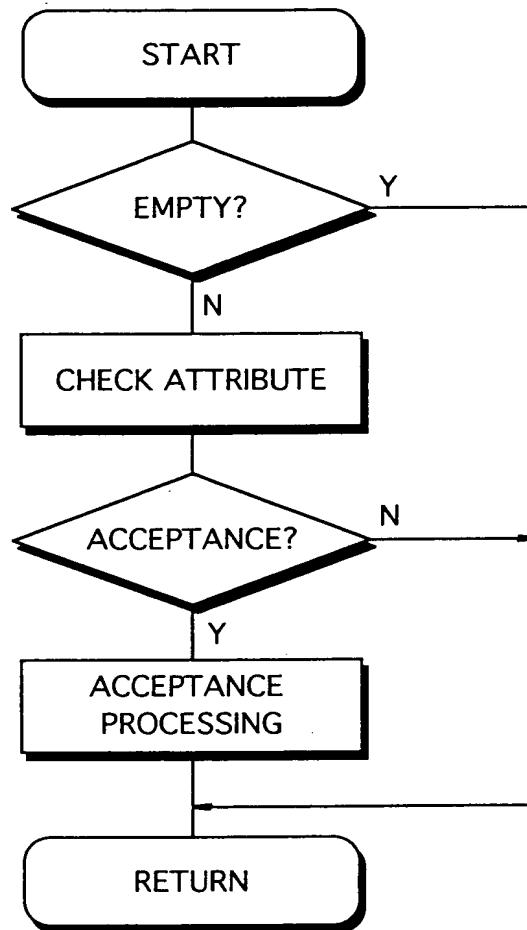
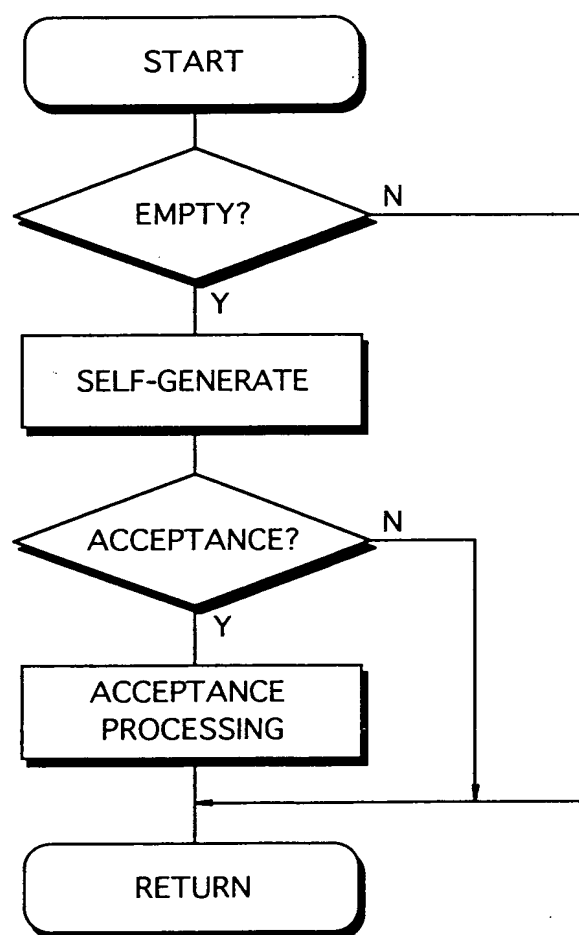


Fig. 17

HOMOGENEITY VECTOR PROGRAM (W02G)



THE **NEW** **YORK** **PUBLIC** **LIBRARY**

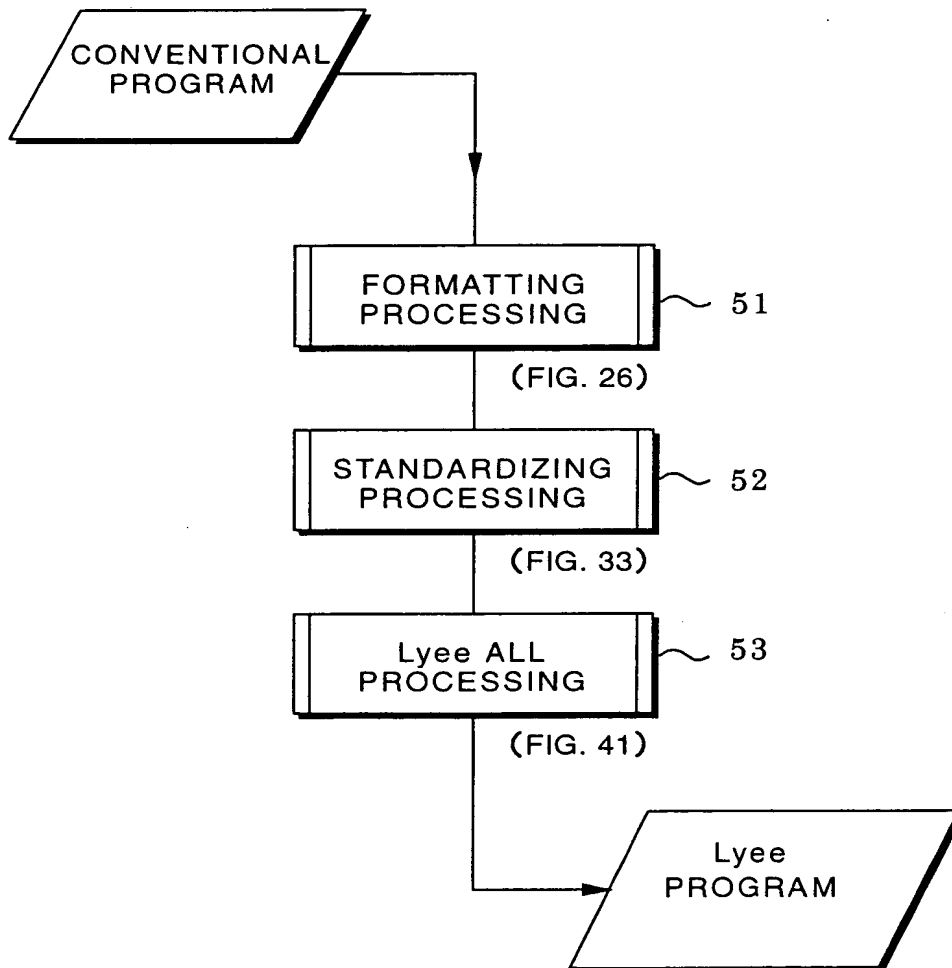
Fig. 18

```

IF W02.□ = LOW-VALUE
    GO TO EXIT.
END-IF
/*CHECK ATTRIBUTE
IF W02.□ = NUMERIC
    GO TO EXIT.
END-IF
IF W02.□ NOT = LOW-VALUE
    GO TO EXIT.
END-IF
W02.□ CNT = W02.□_CNT+1
IF W02.□_CNT < W02_RECALL_MAX
    W02_RECALL_FLG = "1"
ELSE
    W02.□_Non = "1"
END-IF

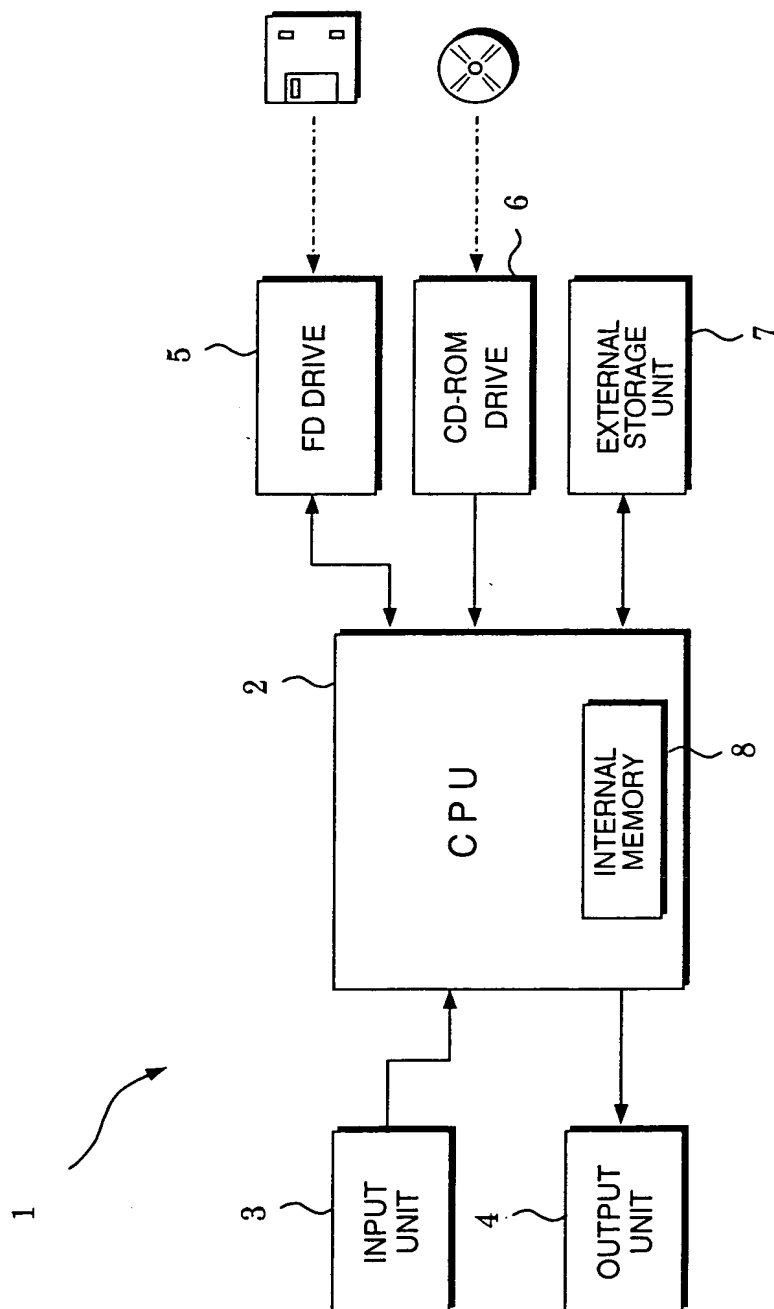
```

19 / 50

Fig. 19

20 / 50

Fig. 20



33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552

Fig. 21

```
01 IDENTIFICATION DIVISION.  
02 PROGRAM-ID. AAA1.  
03 ENVIRONMENT DIVISION.  
04 CONFIGURATION SECTION.  
05     SOURCE-COMPUTER. AS400.  
06     OBJECT-COMPUTER. AS400.  
07 INPUT-OUTPUT SECTION.  
08 FILE-CONTROL.  
09     SELECT GAMEN-F ASSIGN TO SCREEN-AAA.  
10     ORGANIZATION IS TRANSACTION.  
  
11 DATA DIVISION.  
12 FILE SECTION.  
13 FD GAMEN-F.  
14 01 GAMEN-R.  
15     03 SHIN-CD PIC X(05).  
16     03 SU      PIC S9(02).  
17     03 TANKA   PIC S9(05).  
18     03 KINGAKU PIC S9(05).  
  
19 WORKING-STORAGE SECTION.  
20 01 SHIN-TBL.  
21     03 SHIN-CD PIC X(05).  
22     03 TANKA   PIC S9(05).  
23 01 END-BTN PIC X(01).  
24 01 WK      PIC S9(05).  
  
01 PROCEDURE DIVISION.  
02 MAIN-AA SECTION.  
03 MAIN-START.  
04     OPEN I-O GAMEN-F.  
05     INITIALIZE GAMEN-R.  
06     WRITE GAMEN-R.  
07     LOOP-1.  
08     READ GAMEN-F.  
09     IF END-BTN = "1"  
10         CLOSE GAMEN-F  
11         GO TO MAIN-EXIT  
12     END-IF.  
13     IF SHIN-CD OF GAMEN-R = SPACE OR SU OF GAMEN-R = ZERO  
14         MOVE 99999 TO KINGAKU OF GAMEN-R  
15     ELSE  
16         MOVE SHIN-CD OF GAMEN-R TO SHIN-CD OF SHIN-TBL.  
17         SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBLTANKA  
18         IF STATUS NOT = ZERO  
19             MOVE 99999 TO TANKA OF GAMEN-R  
20     ELSE  
21         MOVE TANKA OF SHIN-TBL TO TANKA OF GAMEN-R  
22         COMPUTE WK = TANKA OF GAMEN-R * SU OF GAMEN-R  
23         IF WK > 10000  
24             COMPUTE KINGAKU OF GAMEN-R = WK * 0.8  
25         ELSE  
26             COMPUTE KINGAKU OF GAMEN-R = WK * 0.9  
27         END-IF  
28     END-IF.  
29     END-IF.  
30     WRITE GAMEN-R.  
31     GO TO LOOP-1.  
32 MAIN-EXIT.  
33 STOP RUN.
```

54

PRODUCT ORDERING SCREEN

<PRODUCT ORDERING SCREEN>

PRODUCT CODE AAAAA 55

QUANTITY 20 56

UNIT PRICE 100 57

AMOUNT 2000 58

59 QUIT

POLICE OFFICER

23 / 50

Fig. 23

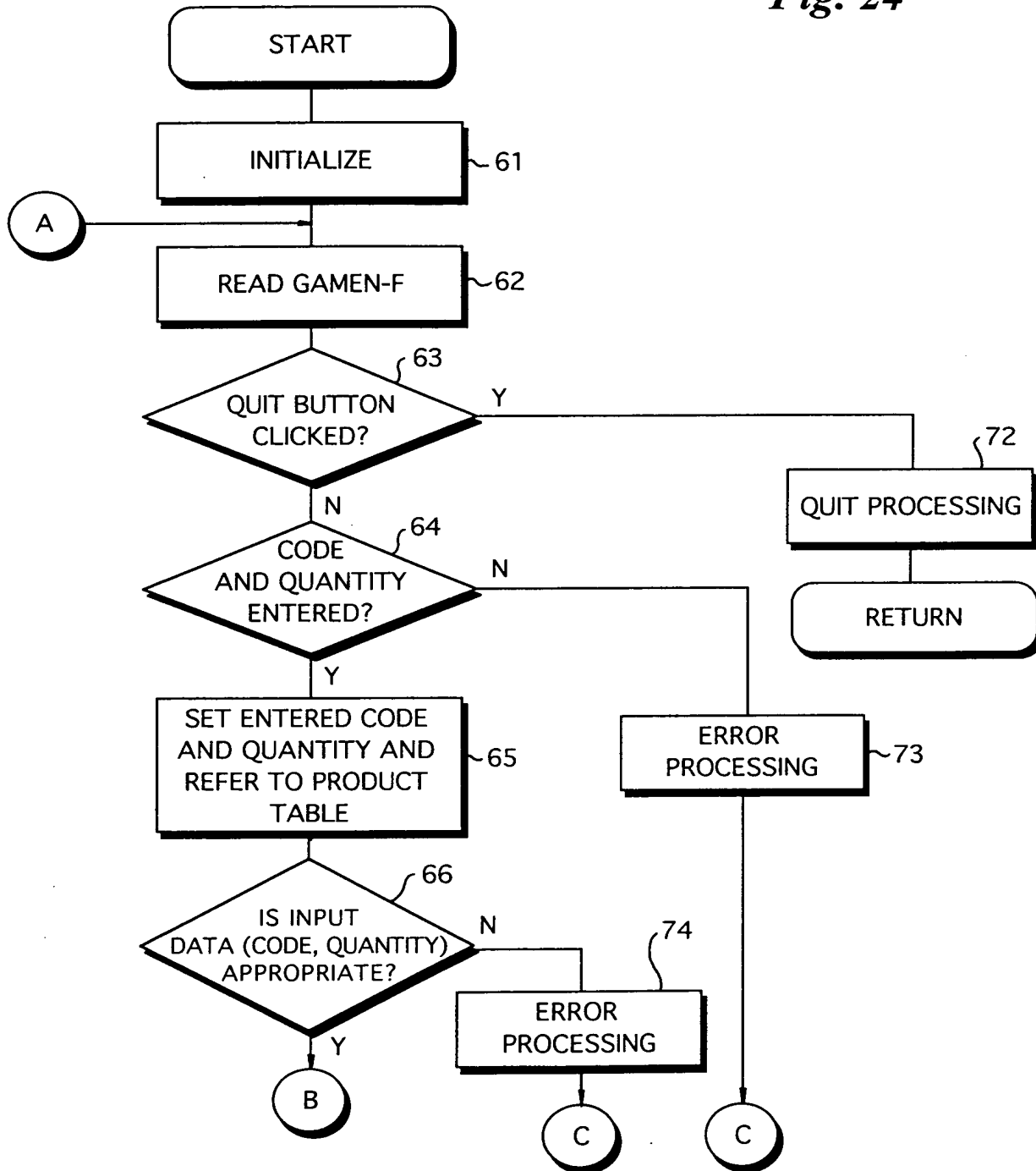
SHIN-TBL (PRODUCT TABLE)

SHIN-CD (PRODUCT CODE)	TANKA (UNIT PRICE)
AAAAA	100
BBBBB	200
CCCCC	300
DDDDD	400
EEEEE	500
⋮	⋮

60

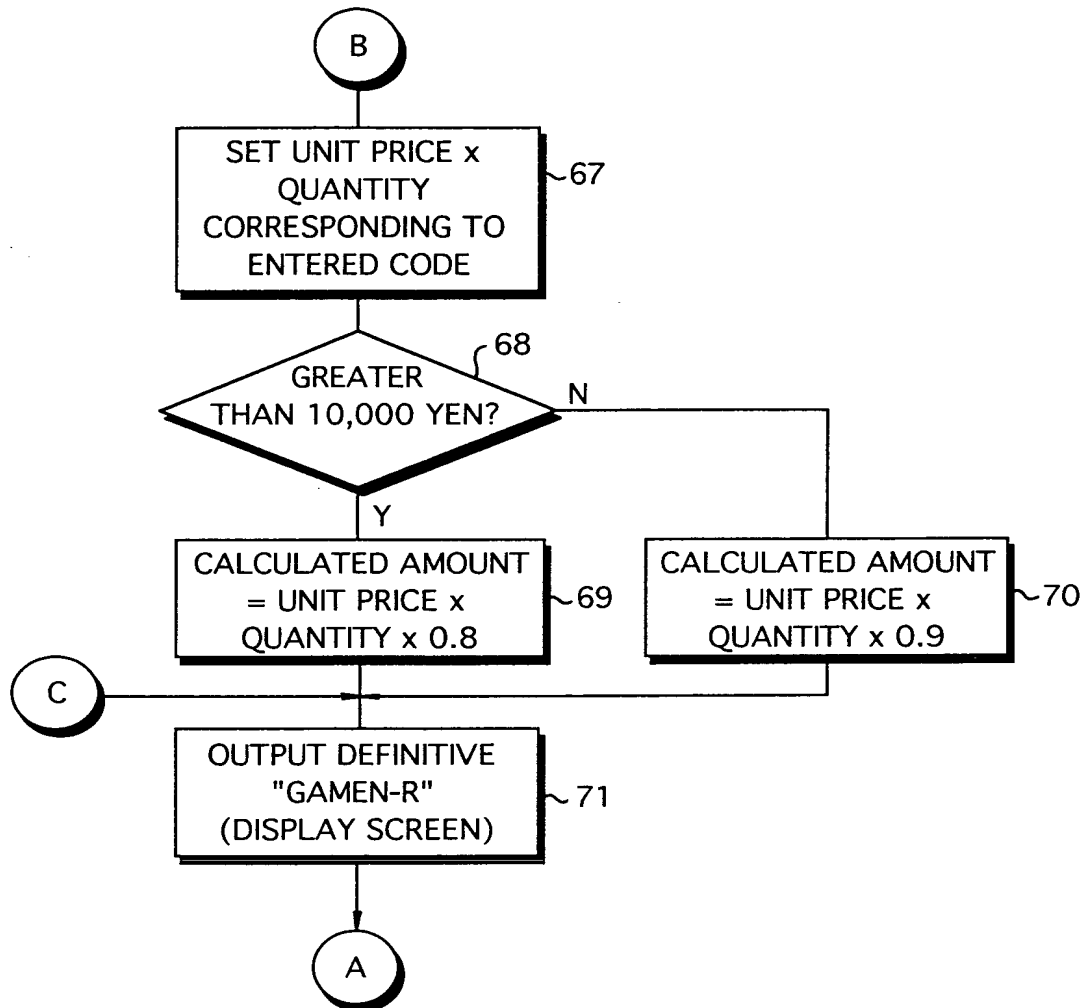
T03040" 00000000

Fig. 24



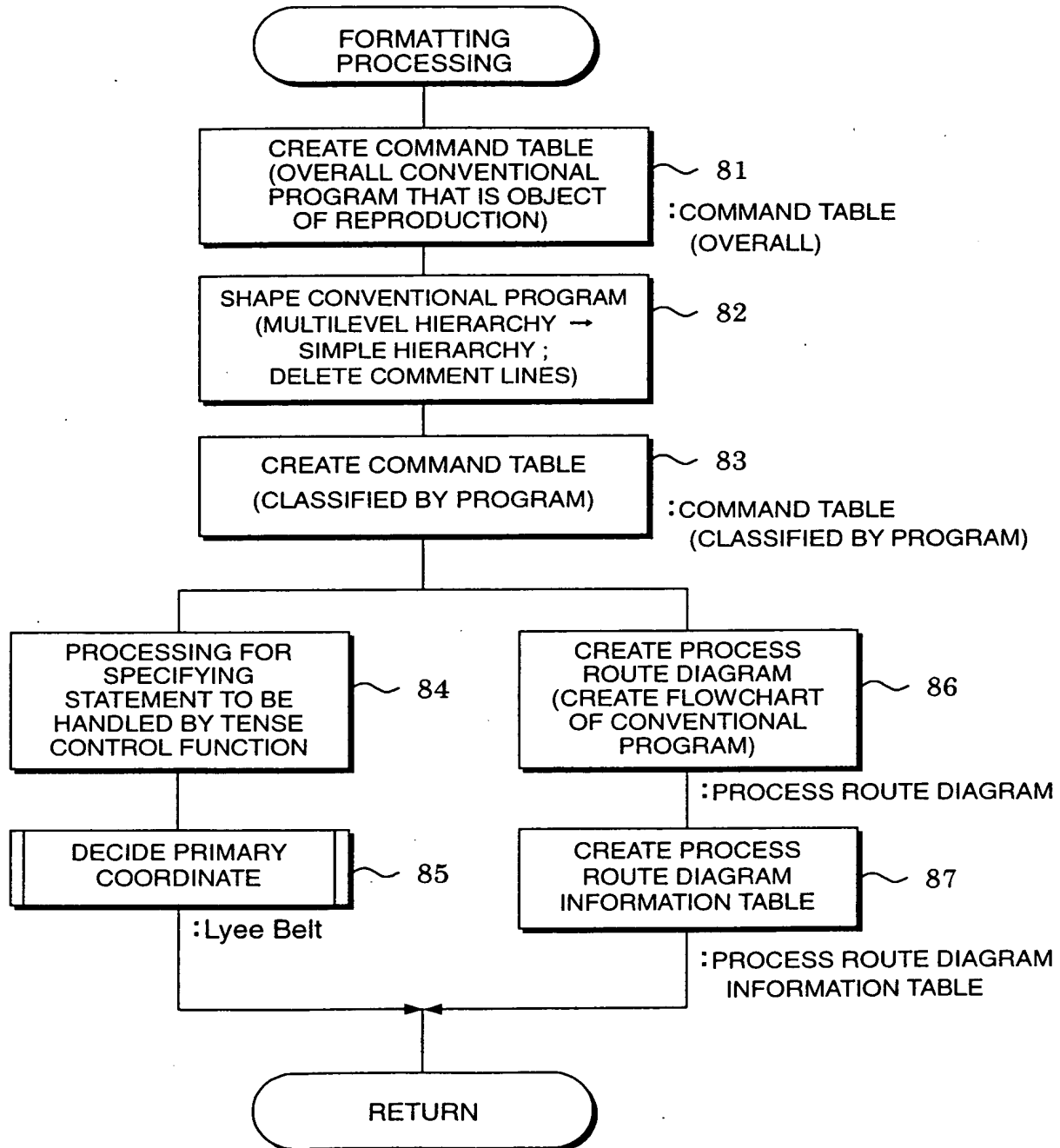
T09020' 0009300

Fig. 25



26 / 50

Fig. 26



T03020" 00000000

09/868893

OBLON ET AL (703) 413-3000

DOCKET # 210P/US/RT SHEET 27 OF 50

09/868893

27/50

Fig. 27

COMMAND TABLE (OVERALL)

DEFINITIVE TYPE	COMMAND	COMMAND TYPE	CONDITION	INPUT/OUTPUT UNIT	INPUT/OUTPUT	PALLET LOCATION
SCREEN	SELECT	COBOL	ASSIGN TO SCREEN	ASSIGN CLAUSE FILE NAME	INPUT	T0W02
SCREEN	SELECT	COBOL	ASSIGN TO SCREEN	ASSIGN CLAUSE FILE NAME	OUTPUT	T0W04
DB	SELECT	COBOL	FROM	FROM CLAUSE DB NAME	INPUT	T1W02

105040" 0000000

Fig. 28

COMMAND TABLE (CLASSIFIED BY PROGRAM)

PROGRAM ID	COMMAND	INPUT/OUTPUT UNIT	DEFINITIVE	INPUT/OUTPUT	PALLET LOCATION
AAA1	READ	AAA	GAMEN-R	INPUT	TOW02
AAA1	WRITE	AAA	GAMEN-R	OUTPUT	TOW04
AAA1	SELECT	SHIN-DB	SHIN-TBL	INPUT	T1W02

28 / 50

29 / 50

Fig. 29

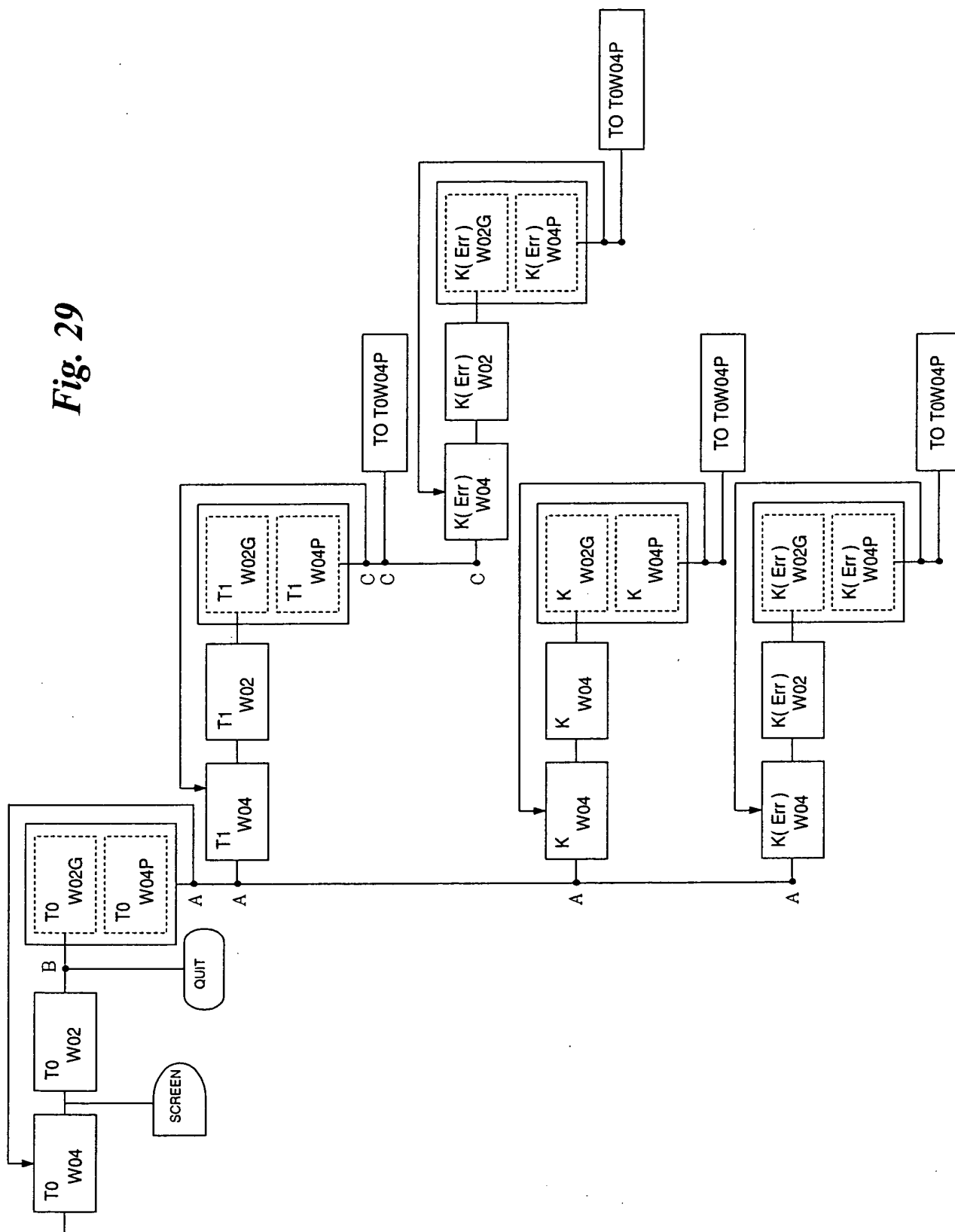


Fig. 30

PROCESS ROUTE DIAGRAM INFORMATION TABLE

PRESENT PGM ID	PROCESS ROUTE DIAGRAM ID	UNIT-TEAM FUNCTION ID	PALLET ID	ROUTING VECTOR ID	NEXT-PALLET ID	INPUT/OUTPUT COMMAND VECTOR ID	COMMAND	DEFINITIVE
A1	Sample1	Sample_1	A1T0W04	Route-1	A1T0W02	WRITE-1	WRITE	GAMEN-R
A1	Sample1	Sample_1	A1T0W02	Route-1	A1T0W03			
				Route-2	STOP			
A1	Sample1	Sample_1	A1T0W03	Route-1	A1T0W04	READ-1	READ	GAMEN-R
				Route-2	A1T1W04			
				Route-3	A1K1W04			
				Route-4	A1E1W04			
A1	Sample2	Sample_2	A1T1W04	Route-1	A1T1W02			
A1	Sample2	Sample_2	A1T1W02	Route-1	A1T1W03			
A1	Sample2	Sample_2	A1T1W03	Route-1	A1T0W04P	READ-1	SELECT	SHIN-DB
				Route-2	A1E2W04			
A1	Sample3	Sample_3	A1K1W04	Route-1	A1K1W02			
A1	Sample3	Sample_3	A1K1W02	Route-1	A1K1W03			
A1	Sample3	Sample_3	A1K1W03	Route-1	A1T0W04P			
A1	Sample4	Sample_4	A1E1W04	Route-1	A1E1W02			
A1	Sample4	Sample_4	A1E1W02	Route-1	A1E1W03			
A1	Sample4	Sample_4	A1E1W03	Route-1	A1T0W04P			
A1	Sample5	Sample_5	A1E2W04	Route-1	A1E2W02			
A1	Sample5	Sample_5	A1E2W02	Route-1	A1E2W03			
A1	Sample5	Sample_5	A1E2W03	Route-1	A1T0W04P			

Fig. 31

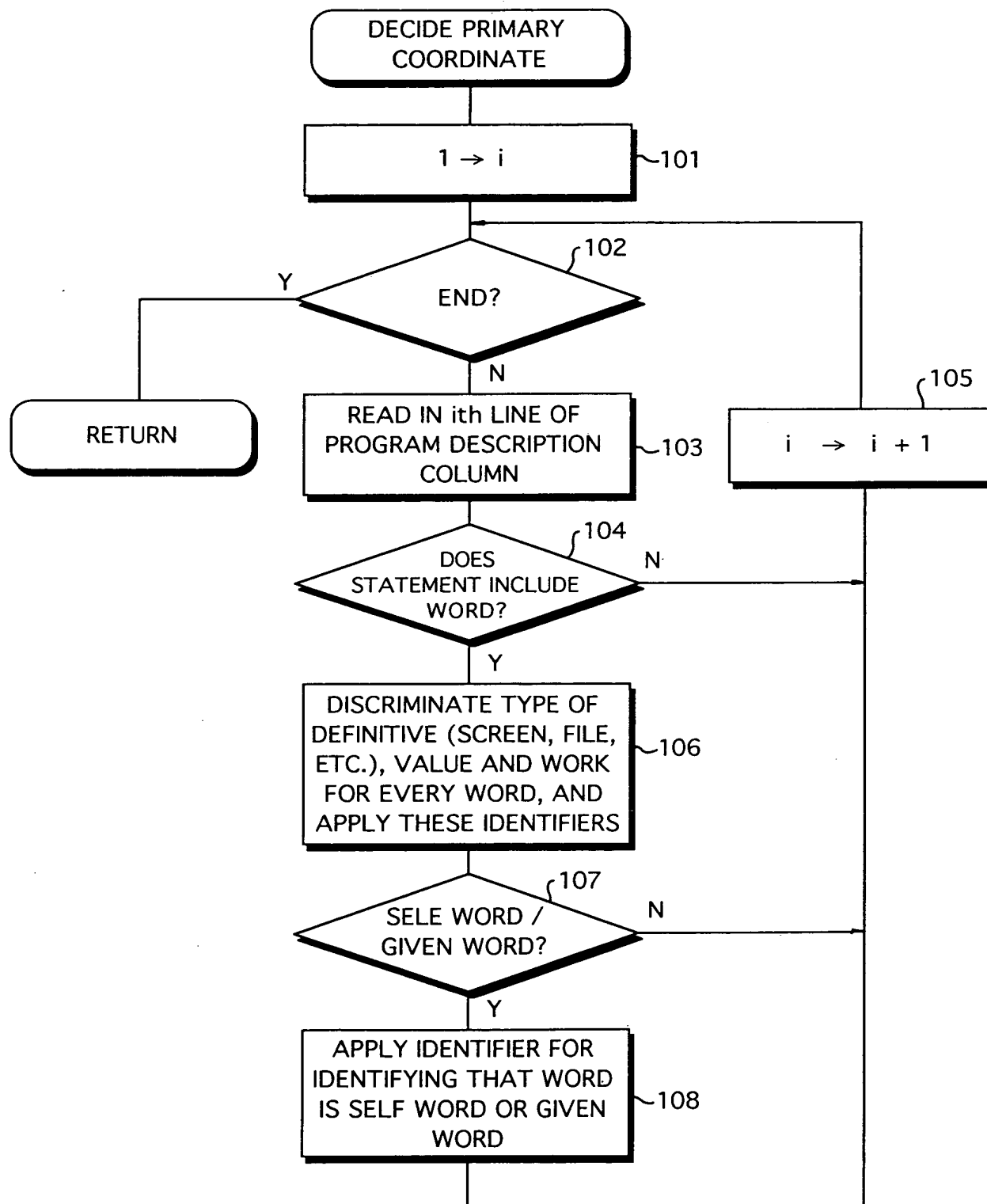


FIG. 31

32/50

Fig. 32

LINE NUMBER	COMMAND TYPE	UNIT-TEAM FUNCTION	PROGRAM DESCRIPTION
01			PROCEDURE DIVISION.
02			MAIN-AA SECTION.
03			MAIN-START.
04		Φ	OPEN I-O GAMEN-F.
05		Φ	INITIALIZE GAMEN-R.
06		Φ	WRITE GAMEN-R.
07			LOOP-1.
08		Φ	READ GAMEN-F.
09	IF		IF END-BTN(SCREEN) = "1"(VALUE)
10		Φ	CLOSE GAMEN-F
11			GO TO MAIN-EXIT
12			END-IF.
13	IF		IF SHIN-CD OF GAMEN-R(SCREEN) = SPACE(VALUE) OR SU OF GAMEN-R(SCREEN) = ZERO(VALUE)
14	SELF COMMAND		MOVE 99999(GIVEN : VALUE) TO KINGAKU OF GAMEN-R(SELF : SCREEN)
15			ELSE
16	SELF COMMAND		MOVE SHIN-CD OF GAMEN-R(GIVEN : SCREEN) TO SHIN-CD OF SHIN-TBL(SELF : CONTROL BOX)
17	CMD		SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBLTANKA
18	IF		IF STATUS(CONTROL BOX) NOT = ZERO(VALUE)
19	SELF COMMAND		MOVE 99999(SELF : VALUE) TO TANKA OF GAMEN-R(SELF : SCREEN)
20			ELSE
21	SELF COMMAND		MOVE TANKA OF SHIN-TBL(GIVEN:DB) TO TANKA OF GAMEN-R(SELF : SCREEN)
22	SELF COMMAND		COMPUTE WK(SELF : WORK)= TANKA OF GAMEN-R(GIVEN : SCREEN) * SU OF GAMEN-R(GIVEN : SCREEN)
23	IF		IF WK(WORK) > 10000(VALUE)
24	SELF COMMAND		COMPUTE KINGAKU OF GAMEN-R(SELF:SCREEN) = WK(GIVEN:WORK) * 0.8
25			ELSE
26			COMPUTE KINGAKU OF GAMEN-R(SELF:SCREEN) = WK(GIVEN:WORK) * 0.9
27			END-IF
28			END-IF
29			END-IF.
30		Φ	WRITE GAMEN-R.
31			GO TO LOOP-1.
32			MAIN-EXIT.
33			STOP RUN.

33 / 50

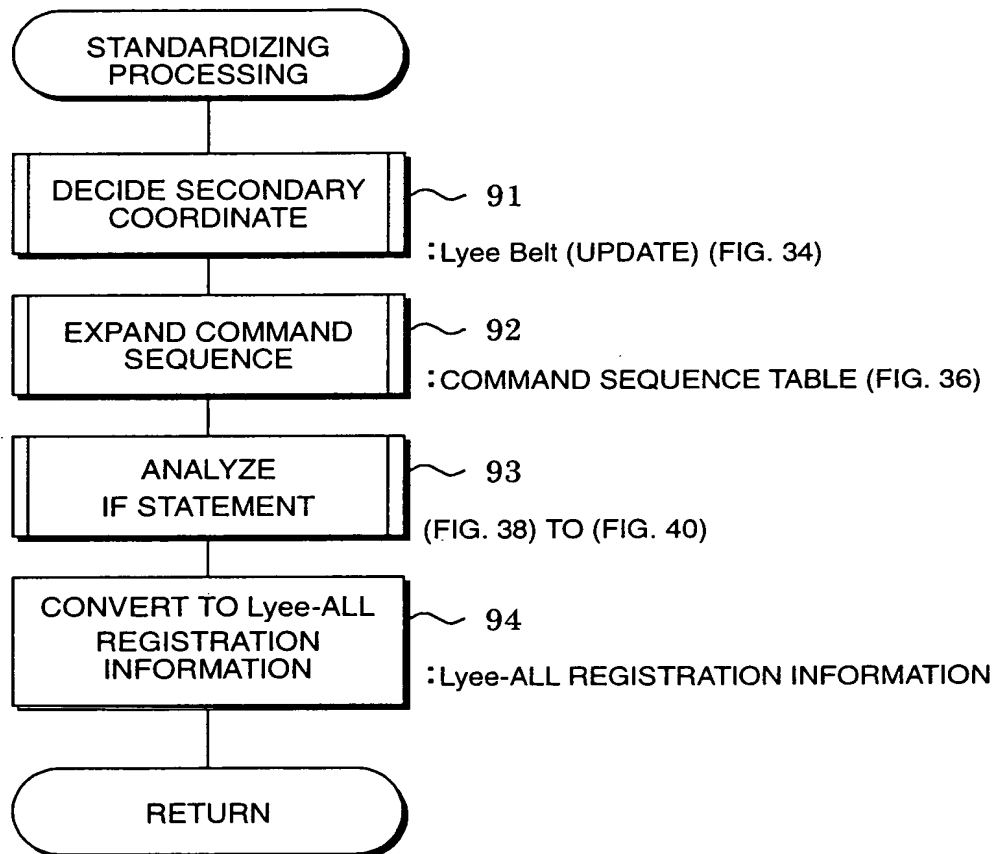
Fig. 33

Fig. 34

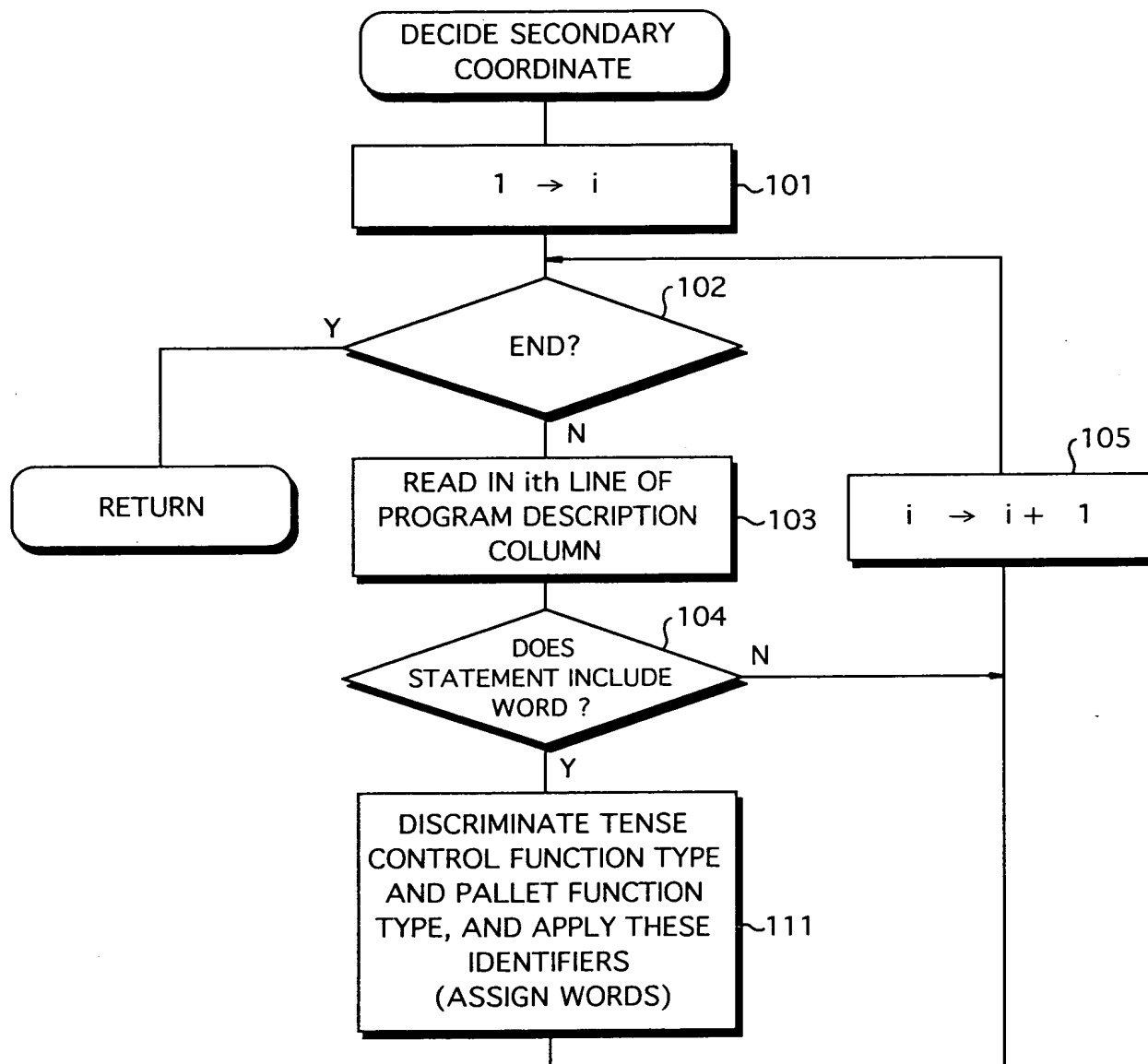


FIG. 34

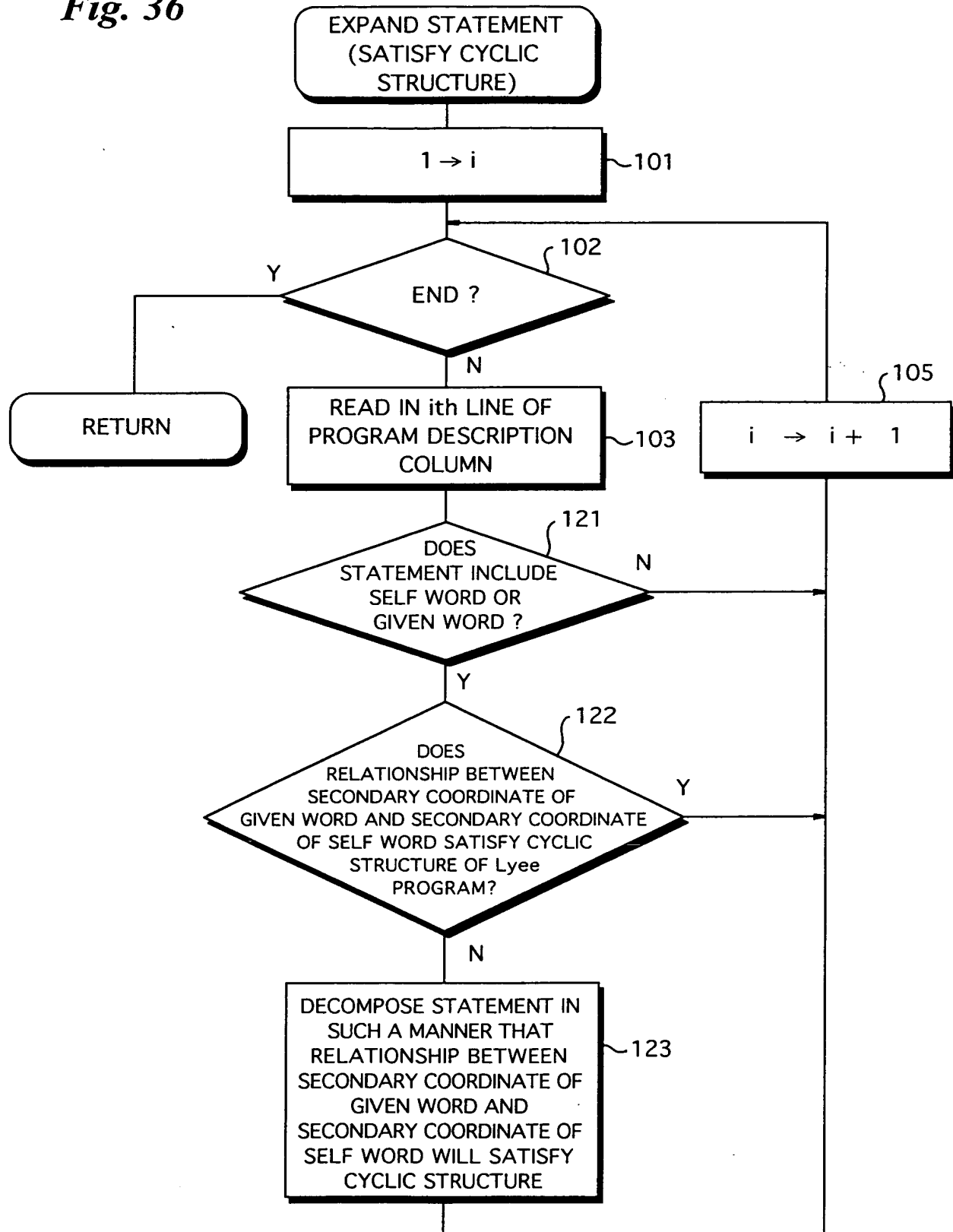
35/50

Fig. 35

LINE NUMBER	COMMAND TYPE	UNIT-TEAM FUNCTION	PROGRAM DESCRIPTION
01			PROCEDURE DIVISION.
02			MAIN-AA SECTION.
03			MAIN-START.
04		Φ	OPEN I-O GAMEN-F.
05		Φ	INITIALIZE GAMEN-R.
06		Φ	WRITE GAMEN-R.
07			LOOP-1:
08		Φ	READ GAMEN-F.
09	IF		IF END-BTN(T0W02) = "1"(VALUE)
10		Φ	CLOSE GAMEN-F
11			GO TO MAIN-EXIT
12			END-IF.
13	IF		IF SHIN-CD OF GAMEN-R(T0W02) = SPACE(VALUE)
14	SELF COMMAND		OR SU OF GAMEN-R(T0W02) = ZERO(VALUE)
15			MOVE 99999(GIVEN : VALUE) TO KINGAKU OF GAMEN-R(SELF:T0W04)
16	SELF COMMAND		ELSE
17	CMD		MOVE SHIN-CD OF GAMEN-R(GIVEN:T0W02)
18	IF		TO SHIN-CD OF SHIN-TBL (SELF:T1CB)
19	SELF COMMAND		SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBL.TANKA
20			IF STATUS(T1CB) NOT = ZERO(VALUE)
21	SELF COMMAND		MOVE 99999(GIVEN:VALUE) TO TANKA OF GAMEN-R(SELF:T0W04)
22	SELF COMMAND		ELSE
23	IF		MOVE TANKA OF SHIN-TBL(GIVEN:T1W02)
24	SELF COMMAND		TO TANKA OF GAMEN-R(SELF:T0W04)
25			COMPUTE WK(SELF:KW04P)= TANKA OF GAMEN-R(GIVEN:T0W04)
26			* SU OF GAMEN-R(GIVEN:T0W02)
27			IF WK(KW04P) > 10000(VALUE)
28			COMPUTE KINGAKU OF GAMEN-R(SELF:T0W04) = WK(GIVEN:KW04P) * 0.8
29			ELSE
30			COMPUTE KINGAKU OF GAMEN-R(SELF:T0W04) = WK(GIVEN:KW04P) * 0.9
31			END-IF
32			END-IF
33		Φ	WRITE GAMEN-R.
			GO TO LOOP-1.
			MAIN-EXIT.
			STOP RUN.

0966093-070604

1. The first part of the paper is devoted to the study of the properties of the function $f(x)$ defined by the equation $f(x) = \int_0^x f(t) dt$. It is shown that $f(x)$ is a continuous function and that it satisfies the functional equation $f(x+y) = f(x) + f(y)$. The function $f(x)$ is also shown to be differentiable and its derivative is found to be $f'(x) = f(x)$. This implies that $f(x) = Ce^x$ for some constant C . The value of C is determined by the initial condition $f(0) = 1$, which gives $C = 1$. Therefore, the function $f(x)$ is $f(x) = e^x$.



37/50

Fig. 37

COMMAND SEQUENCE TABLE

LINE NUMBER	TYPE	LOCATION	STATEMENT AFTER EXPANSION
21	DUPLICATE VECTOR	T1W02G	MOVE TANKA OF SHIN-TBL(T1W02) TO TANKA OF SHIN-TBL(T1W02G)
21	DUPLICATE VECTOR-TYPE HOMOGENEITY VECTOR	T1W04P	MOVE TANKA OF SHIN-TBL(T1W02G) TO TANKA OF SHIN-TBL(T1W04P BOUNDARY)
21	R-TYPE DUPLICATE VECTOR	T0W04P	MOVE TANKA OF SHIN-TBL(T1W04P BOUNDARY) TO TANKA OF SHIN-TBL(T0W04P BOUNDARY)
21	HOMOGENEITY VECTOR	T0W04P	MOVE TANKA OF SHIN-TBL(T0W04P BOUNDARY) MOVE TANKA TO TANKA OF GAMES-R(T0W04P)
21	DUPLICATE VECTOR	T0W04	MOVE TANKA OF GAMES-R(T0W04P) TO TANKA OF GAMES-R(T0W04)

Fig. 38

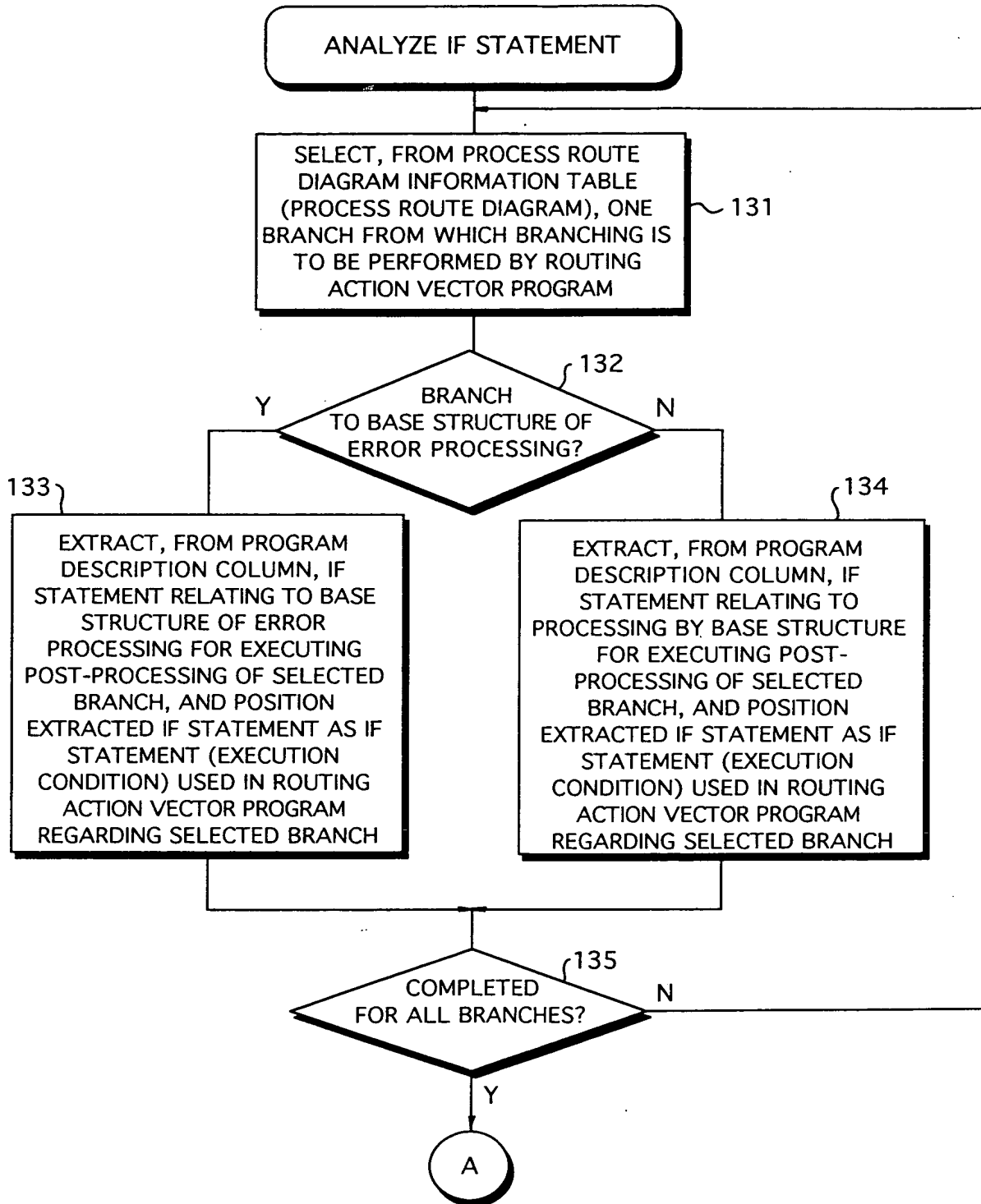


Fig. 39

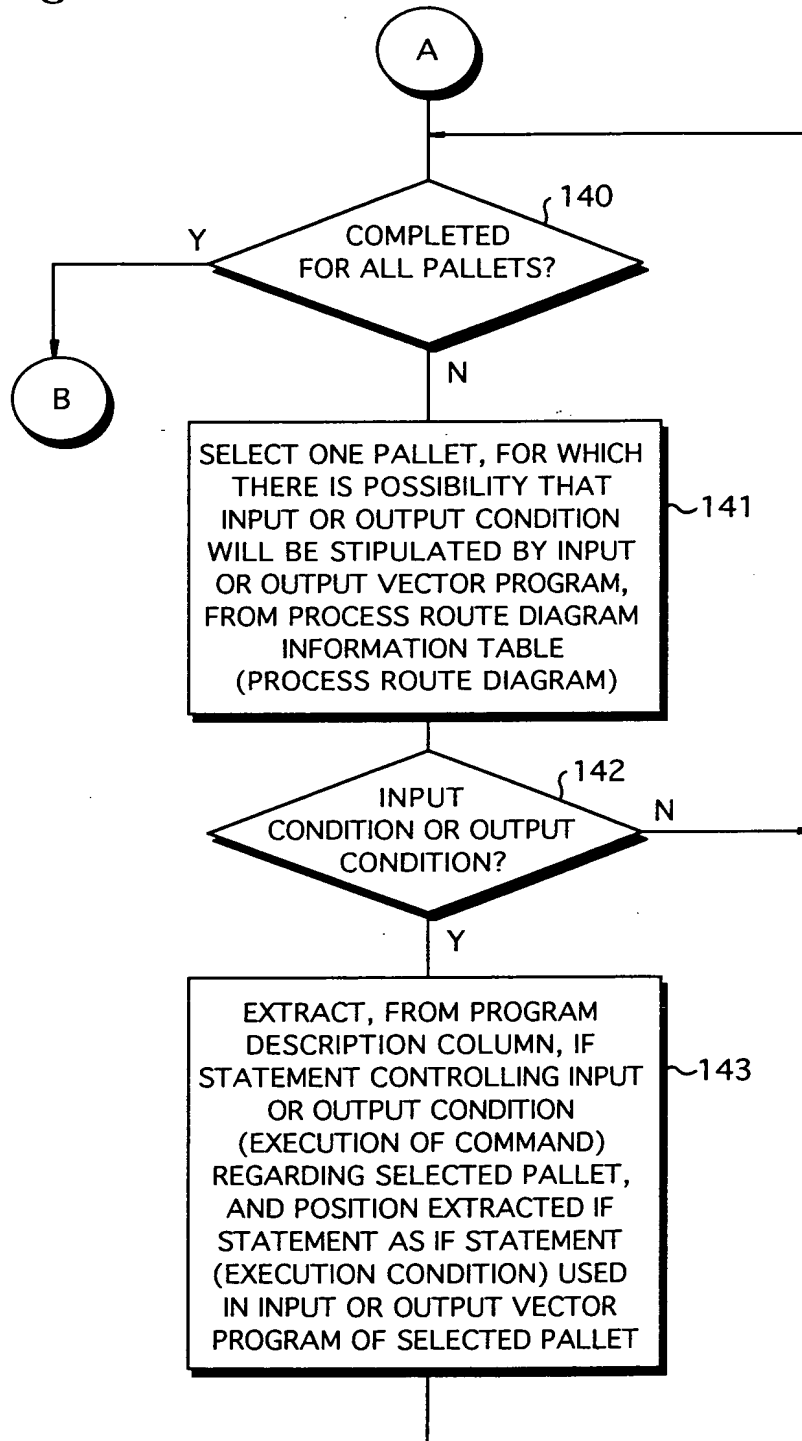
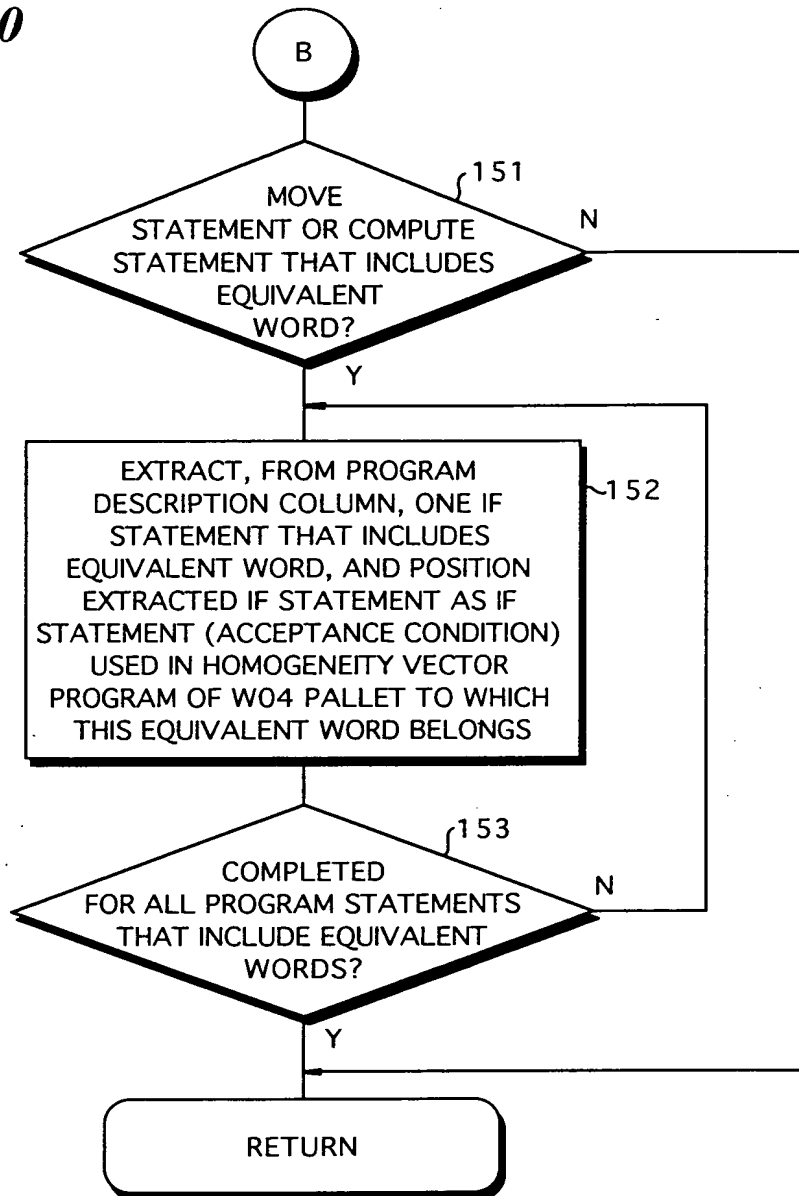
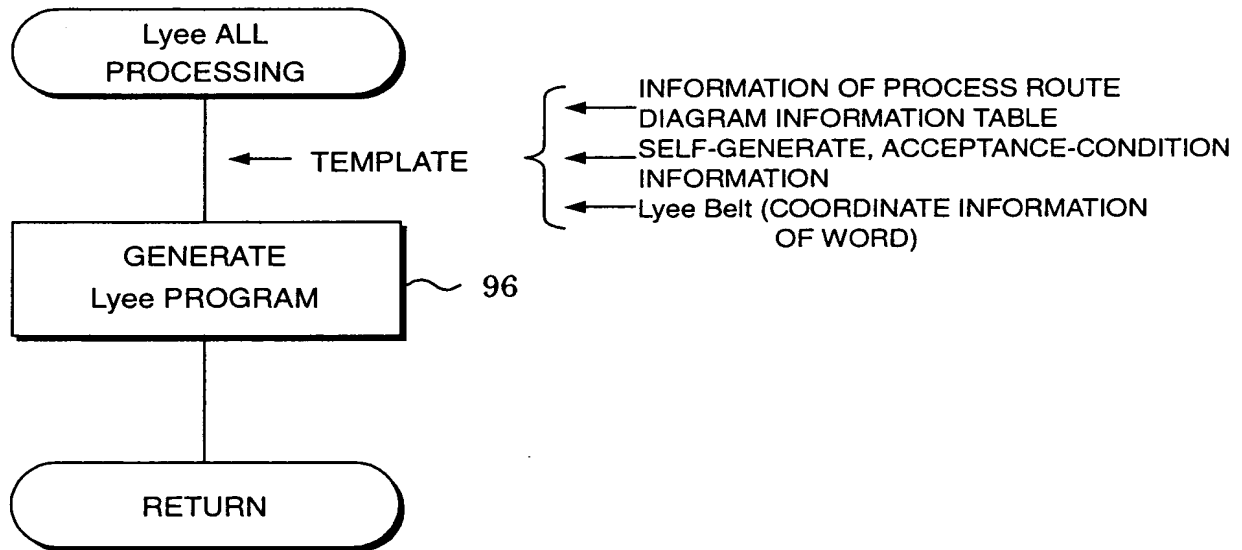


Fig. 40



41 / 50

Fig. 41



42 / 50

Fig. 42

BEFORE EXPANSION

WORKING-STORAGE
SECTION.

COPY A.

01 B-DATA.
COPY B.

A PROC

01 A.

03 A1 PIC X(5).
03 A2 PIC X(10).

END

B PROC

03 B1 PIC X(3).
03 B2.05 B3 PIC X(2).
05 B4 PIC X(5).

END

AFTER EXPANSION

WORKING-STORAGE
SECTION.

01 A.

03 A1 PIC X(5).
03 A2 PIC X(10).

01 B-DATA.

03 B1 PIC X(3).

03 B2.

05 B3 PIC X(2).
05 B4 PIC X(5).

T05020" 2688893

43 / 50

Fig. 43

BEFORE EXPANSION

PROCEDURE	DIVISION.
MOVE A TO B.	
COPY X.	
MOVE C TO D.	

X PROC
IF B > 100
CALL'SUB001'USING B C
ELSE
CALL'SUB002'USING B C.
END



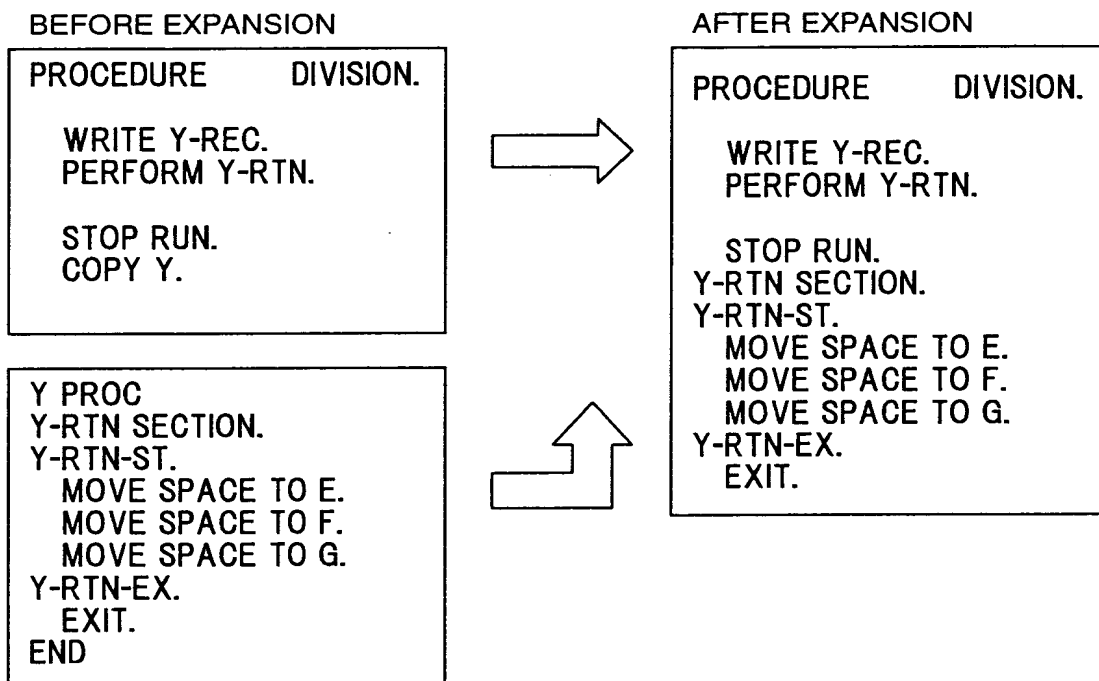
AFTER EXPANSION

PROCEDURE	DIVISION.
MOVE A TO B.	
IF B > 100	
CALL'SUB001'USING B C	
ELSE	
CALL'SUB002'USING B C.	
MOVE C TO D.	



09/868893

44 / 50

Fig. 44

T05020" E0003000

Fig. 45

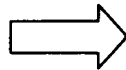
BEFORE EXPANSION

MAIN PROGRAM

```

WORKING-STORAGE SECTION.
01 WK-A      PIC 9(10).
01 WK-B      PIC 9(10).
01 WK-C      PIC 9(10).
....
PROCEDURE    DIVISION.
:
:
MOVE A TO WK-A.
MOVE B TO WK-B.
CALL 'SUB001'
      USING WK-A WK-B WK-C.
MOVE WK-C TO C.
:

```



AFTER EXPANSION

MAIN PROGRAM

```

WORKING-STORAGE SECTION.
01 WK-A      PIC 9(10).
01 WK-B      PIC 9(10).
01 WK-C      PIC 9(10).
....
PROCEDURE    DIVISION.
:
:
MOVE A TO WK-A.
MOVE B TO WK-B.
WK-C=2 * WK-A * WK-B *
WK-B
:
MOVE WK-C TO C.
:

```

SUBROUTINE

```

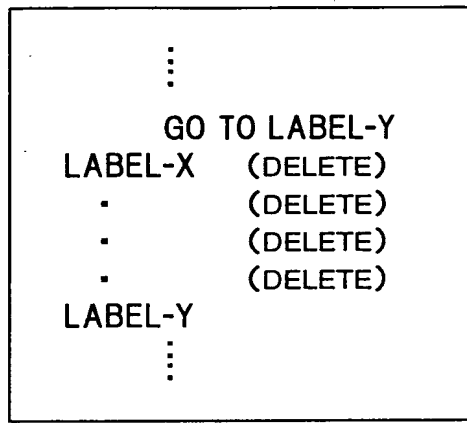
PROGRAM-ID.    SUB001.
....
LINKAGE SECTION.
01 WK-X      PIC 9(10).
01 WK-Y      PIC 9(10).
01 WK-Z      PIC 9(10).
PROCEDURE    DIVISION.
      USING WK-X WK-Y WK-Z.
PROC-RTN.    SECTION.
PROC-ST.

      COMPUTE
        WK-Z=2 * WK-X * WK-Y *
WK-Y
PROC-EX.
      EXIT PROGRAM.

```

105020" E688893

46 / 50

Fig. 46

FBI/DOJ E688893

47 / 50

Fig. 47

SET IDX TO 1 \Rightarrow MOVE 1 TO IDX

09/868893 070604

48 / 50

Fig. 48

ORIGINAL COMMAND	AFTER SHAPING
MOVE A TO B,C ⇒	MOVE A TO B MOVE A TO C

T05020" C009800

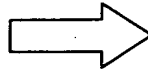
Fig. 49

ORIGINAL COMMAND

```

IF      A = B
  AND C = D
  IF    X = Y
    AND Z = W
      MOVE E TO
F
ELSE
  MOVE G TO H

```



AFTER SHAPING

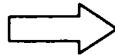
```
IF A = B AND C = D
IF X = Y AND Z = W
MOVE E TO F
ELSE
MOVE G TO H
END-IF
ELSE
END-IF
```

1. The first part of the report discusses the importance of maintaining accurate records of all transactions and the role of the auditor in ensuring the integrity of the financial statements.

50 / 50

*Fig. 50*BEFORE CONVERSION
TO SIMPLE HIERARCHY

```
IF A = B
  PERFORM S1
ELSE
  PERFORM S2
  ...
S1 SECTION
  CONTENT OF S1
EXIT
  ...
S2 SECTION
  CONTENT OF S2
EXIT
```

AFTER CONVERSION
TO SIMPLE HIERARCHY

```
IF A = B
  CONTENT OF S1
ELSE
  CONTENT OF S2
```

T03020" E6889360